AVARICHY Mobile Application :

This ZIP file contains **the data, code, video, and presentation** for our innovative solution Avarichy, a mobile application built to support financial operations. Developed with React Native for the frontend and Node.js (Express) with MongoDB for backend services, this application integrates with four fine-tuned AI models via Gemini 1.5 Flash APIs. The codebase implements the CRISP-DM methodology for data processing and includes Jupyter notebooks documenting data extraction and preprocessing pipelines in both English and Arabic, with additional support for speech capabilities using Hugging Face models.

Project Structure:

-The pptx file contains then presentation of our solution

-The video contains the demo of our mobile application

/Data Folder contains:

- **Datasets**: Used for fine-tuning our four models
- **PDF File**: Used for parsing

/Code Folder contains:

1. Mobile Application (/mobile_app) This directory contains all code related to the mobile app and its backend:

- Frontend: Developed using React Native, providing a user interface for financial interactions and model insights
    - Contains React Native components, screens, and navigation structure
- Backend: Built with Node.js (Express) and MongoDB for handling user data, session management, and API requests
    - Manages user and session data handling, model requests, and response processing
- Gemini API Integration: Interacts with four fine-tuned financial models using Gemini 1.5 Flash APIs
    - Includes routes and controllers for managing API calls to the Gemini models

2. CRISP-DM Implementation (/crisp_dm) This folder implements the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology CRISP-DM Code: Contains Python

scripts implementing the methodology, focusing on data Extraction,cleaning and transformation

    - Location: crisp_dm/

3. Data Processing and Extraction Notebooks (/Crispdm folder) Contains two Jupyter notebooks detailing data extraction and preprocessing processes for both English and Arabic datasets (Arabic dataset translated from English):

- Data Extraction: Used to parse and extract data from PDF files

    - Location: notebook/data_Extraction_And_Preprocessing
    - Documents the process of extracting structured financial data from PDFs and data generated using prompt engineering
- Data Preparation and Preprocessing: Python-based preprocessing of English data, with translation and preparation of Arabic data for fine-tuning both versions
- Hugging Face Speech Models: Integrates speech-to-text and text-to-speech models, with testing documentation
    - Location: notebook/
    - Documents tests and results for speech-to-text and text-to-speech conversions

Configuration and API Keys (/)

- API Key Management:
    - File: api_keys.txt – Stores API keys required for the four fine-tuned models

Technology Stack

- Frontend: React Native
- Backend: Node.js, Express
- Database: MongoDB
- AI Models: Four fine-tuned models using Gemini 1.5 Flash APIs
- Speech Models: Hugging Face models for speech-to-text and text-to-speech
- Data Processing: Python (for extraction and preprocessing)